Efficient Active SLAM based on Submap Joining

Yongbo Chen[1,2], **Shoudong Huang**[1], **Robert Fitch**[1], **Jianqiao Yu**[2]
1. Centre for Autonomous Systems, University of Technology Sydney, Sydney, Australia
2. School of Aerospace Engineering, Beijing Institute of Technology, Beijing, China
Yongbo.Chen@uts.student.edu.au

Abstract

This paper considers the active SLAM problem where a robot is required to cover a given area while at the same time performing simultaneous localization and mapping (SLAM) for understanding the environment and localizing the robot itself. We propose a model predictive control (MPC) framework, and the minimization of uncertainty in SLAM and coverage problems are solved respectively by the Sequential Quadratic Programming (SQP) method. Then, a decision making process is used to control the switching of two control inputs. In order to reduce the estimation and planning time, we use Linear SLAM, which is a submap joining approach. Simulation results are presented to validate the effectiveness of the proposed active SLAM strategy.

1 Introduction

Active SLAM problem has been approached in the past as an action selection problem in order to improve SLAM results and also perform other tasks such as coverage or exploration. It is considered as one of the most challenging problems of mobile robotics in an unknown environment [Cadena, 2016]. It presents a well-known dilemma for the activity of the robot, how to strike a balance between visiting new places and revisiting known areas to obtain a good map.

Several known approaches for active SLAM include the MPC framework [Huang, 2005] and the partially observably Markov decision process formalism [Kaelbling, 1998]. In order to select the best future action from a set of alternatives, it entails two issues, the computation of a cost function to evaluate the effect of each candidate action, and the process of seleting the optimal action set. In order to evaluate the SLAM results, the information matrices of the estimated vector after executing the future actions are normally used. Comparing a certain metric of the information matrices is a direct idea of evaluating actions in terms of estimation accuracy. The Theory of Optimal Experimental Design (TOED) [Pazman, 1986], including *A-opt*, *D-opt*, and *E-opt*, is commonly used in active SLAM. In [Carrillo, 2012] and [Carrillo, 2015], a comparison of these optimality criteria is performed and it is shown that only *D-opt* retains monotonicity during the exploration phase of an active SLAM algorithm for the linearized framework. Thus, the *D-opt* metric, which is to maximize the determinant of the information matrix, is the best criterion to use in most situations.

Because of the unknown future measurements, obtaining future information matrices accurately is difficult. The main idea of solving this problem is to introduce a assumption for these measurements, such as the zero-innovation measurement assumption and random measurement assumption. In [Leung, 2006], the future information matrix is obtained by zero-innovation extended Kalman filter (EKF) and extended Information Filter (EIF) prediction. Expectation-maximization (EM) and a Gauss-Newton (GN) approach are applied together in [Indelman, 2015] to solve this problem with a random measurement assumption. In [Indelman, 2015], a conservative sparse information space is used to reduce the computational complexity for computing the predicted *D-opt* objective function for the candidate action.

As an optimal control problem, to find the global optimal solution for selecting the best future action is usually hard. Some researchers choose future waypoints from a small subset of locations to reduce the size of the search space, for example, frontier-based exploration in [Matan, 2014]. This approach changes the planning problem into the discrete optimization domain. Some approaches can only find locally optimal policies in continuous-space planning under uncertainty [Berg, 2012]. Active SLAM has also been addressed using path planning algorithms such as RRT* [Vallvé, 2015], D* [Maurović, 2017], and potential information fields [Vallvé, 2015]. In genereal, finding an optimal solution for active SLAM is still an open problem.

In this paper, the MPC method and submap joining are applied to greatly reduce the problem difficulty and the planning time. The objective functions employed include the *D-opt* optimality criterion and area coverage taking into account the localization uncertainties. For the *D-opt* MPC problem, the prediction of the future information matrix is obtained based on the assumption of the perfect (zero-innovation) measurement. For the coverage problem, considering the pose uncertainty, we formulate it as an optimization problem in the MPC framework. The Linear SLAM submap joining approach [Zhao, 2013] is used in the SLAM process in order to improve run-time performance. Simulations results are presented to validate the proposed algorithm.

The main contributions of this work are: (1) the use of submap joining in active SLAM (Section 4), which greatly improves the algorithm's running-time ability, and (2) a new formulation of area coverage considering the robot localization uncertainty, especially when submap joining based SLAM is used (Section 3.3 and 4).

The paper is organized as follows. Section 2 states the motion and observation model and the active SLAM problem. Section 3 describes the MPC strategy and the objective functions for uncertainty minimization and coverage under localization uncertainties. Section 4 discusses the issues related to using submap joining in active SLAM. Simulation results are presented in Section 5. Finally, Section 6 concludes the paper.

2 Model and Problem Statement

2.1 Vehicle Motion Model and Observation Model

The kinematic equations of a robot moving in a 2D environment can be written as:

$$\begin{gathered}\begin{bmatrix} V\Delta T \\ 0 \end{bmatrix} = R_k^T(\boldsymbol{x}_{k+1}^v - \boldsymbol{x}_k^v) + \begin{bmatrix} \delta x_k \\ \delta y_k \end{bmatrix}, \\ \omega_k = \frac{\theta_{k+1}-\theta_k}{\Delta T} - \delta\omega_k \end{gathered} \tag{1}$$

where ΔT is a discrete time interval, $(\boldsymbol{x}_k^{vT}, \theta_k) = (x_k, y_k, \theta_k)$ is the pose of the robot at the k-th step, $R_k = \begin{pmatrix} \cos(\theta_k) & -\sin(\theta_k) \\ \sin(\theta_k) & \cos(\theta_k) \end{pmatrix} \in SO(2)$ is the rotation matrix of the k-th pose of the robot, ω_k is the angular velocity of the robot at the k-th step. V is the velocity which is assumed to be constant. δx_k, δy_k and $\delta\omega_k$ are the discrete time noises of the coordinates and angular velocity and are assumed to be zero-mean Gaussian, meeting $\delta x_k \sim \mathcal{N}(0, \delta_{vx}(k)^2)$, $\delta y_k \sim \mathcal{N}(0, \delta_{vy}(k)^2)$ and $\omega_k \sim \mathcal{N}(0, \delta_{\omega}(k)^2)$.

The observation model is given by:

$$\boldsymbol{Z}_k^{fi} = R_k^T(\boldsymbol{x}^{fi} - \boldsymbol{x}_k^v) + \begin{bmatrix} w_x^k \\ w_y^k \end{bmatrix}, \tag{2}$$

where $\boldsymbol{Z}_k^{fi}$ is the observed value of the i-th feature at k-th step, $\boldsymbol{x}^{fi}$ is the coordinate of the i-th feature, w_x^k, w_y^k are the noises of the sensor in the x and y axes and are assumed to be zero-mean Gaussian, meeting $w_x^k \sim \mathcal{N}(0, \delta_{fx}(k)^2)$ and $w_y^k \sim \mathcal{N}(0, \delta_{fy}(k)^2)$.

2.2 Problem Statement

The active SLAM problem considered in this paper is to allow a robot to select optimal/sub-optimal control inputs to perform two specific tasks (uncertainty minimization and coverage). The environment is unknown except the boundary of the area of interest to be covered. The environment is assumed to contain a number of point features with unknown locations. The robot can observe features within its sensor range. The goal is to cover the area of interest as quickly as possible, while performing 2D point feature based SLAM continuously with accurate SLAM results.

3 MPC Framework

In most situations, the uncertainty minimization task and the coverage task will lead to conflicting robot actions. The uncertainty minimization task will lead the robot to visit known space to obtain more observations; while the coverage task will lead the robot to explore new areas and thus resulting in large uncertainty in SLAM result. We now describe the MPC framework for considering these tasks jointly.

3.1 MPC for Uncertainty Minimization Task

Consider the uncertainty minimization problem under the MPC framework with L-step look-ahead. The objective function is minimized over the time horizon $[k, k+L]$. Similar to [Indelman, 2015], the objective function of the uncertainty minimization task is based on the generalized belief at the L-th planning step:

$$\begin{aligned} & J_k(u_{k:k+L-1}) = f_J(gb(\boldsymbol{X}_{k+L})) = -log\left(det(I_{k+L}(\boldsymbol{X}_{k+L}^{opt}))\right) \\ & \boldsymbol{X}_{k+L}^{opt} = \underset{\boldsymbol{X}_{k+L}}{argmin}\ -log(p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L), \boldsymbol{Z}_{k+1:k+L}) \\ & \boldsymbol{B}(k,L) = \boldsymbol{Z}_{1:k}, u_{0:k-1}, u_{k:k+L-1} \end{aligned}, \tag{3}$$

where $u_{0:k+L-1}$ denotes the $k+L$ control inputs $u_j = \omega_j$, $j = 0, \cdots, k+L-1$, $\boldsymbol{X}_{k+L}$ is the real state vector including the $k+L-1$ poses and the coordinate of the features, $gb(\boldsymbol{X}_{k+L})$ is the Gaussian belief of $\boldsymbol{X}_{k+L}$, $gb(\boldsymbol{X}_{k+L}) \approx \mathcal{N}(\boldsymbol{X}_{k+L}^{opt}, I_{k+L}(\boldsymbol{X}_{k+L}^{opt})^{-1})$, $\boldsymbol{Z}_{1:k}$ and $\boldsymbol{Z}_{k+1:k+L}$ denote the observed values from step 1 to k and step $k+1$ to $k+L$, respectively.

According to Bayes' theorem, we have:

$$\begin{aligned} & p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ & = \frac{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{B}(k,L))p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L))}{p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{B}(k,L))}. \end{aligned} \tag{4}$$

Similar to [Indelman, 2015], assuming that the prior $p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{B}(k,L))$ is uninformative. Then (5) can be rewritten as:

$$\begin{aligned} & p(\boldsymbol{X}_{k+L}|\boldsymbol{Z}_{1:k}, u_{0:k-1}, \boldsymbol{Z}_{k+1:k+L}, u_{k:k+L-1}) \\ & \propto p(\boldsymbol{Z}_{k+1:k+L}|\boldsymbol{X}_{k+L}, \boldsymbol{B}(k,L))p(\boldsymbol{X}_{k+L}|\boldsymbol{B}(k,L)) \end{aligned}. \tag{5}$$

Regarding the future measurement $\boldsymbol{Z}_{k+1:k+L}$, it is an unknown and probabilistic event. If the j-th feature is outside the sensor range from the actual position of the robot, it will not be observed. Since the exact value of $\boldsymbol{Z}_{k+1:k+L}$ is unknown until the actual measuremt is taken, we assume the measurement $\boldsymbol{Z}_{k+1:k+L}$ is perfect (zero-innovation) in the planning step. Using the Markov property, we obtain the predicted future state vector $\boldsymbol{X}_{k+L}^{opt}$:

$$\begin{aligned}\boldsymbol{X}_{k+L}^{opt} =& \underset{\boldsymbol{X}_{k+L}}{argmin} \\ -log(p(\boldsymbol{X}_k|\boldsymbol{Z}_{1:k}, u_{0:k-1})&p(\boldsymbol{X}_{k:k+L}|\boldsymbol{X}_k, u_{k:k+L-1}))\end{aligned} \quad . \tag{6}$$

The first part of (6) is a classical SLAM problem, and the second is a prediction process with zero-innovation probabilistic measurement $\boldsymbol{Z}_{k+1:k+L}$. Assuming that the SLAM result is $\boldsymbol{X}_k^{opt}$, the predicted future pose $\boldsymbol{X}_{k+L}^{opt}$ will be:

$$\boldsymbol{X}_{k+L}^{opt} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ \boldsymbol{x}_{k+1}^{opt} \\ \boldsymbol{x}_{k+2}^{opt} \\ \vdots \\ \boldsymbol{x}_{k+L}^{opt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_k^{opt} \\ f_v(\boldsymbol{x}_k^{opt}, u_k) \\ f_v(\boldsymbol{x}_{k+1}^{opt}, u_{k+1}) \\ \vdots \\ f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1}) \end{bmatrix}, \tag{7}$$

where $f_v(\star)$ is the motion equation shown in (1) and $\boldsymbol{x}_i^{opt}, i = 1, 2, ..., k+L$ is the predicted pose at step i.

The optimization problem for uncertainty minimization using the *D-opt* optimality criterion is:

$$\begin{aligned} &\min_{u_{k:k+L-1}} \quad f_a(u_{k:k+L-1}) = -log(det(I_{k+L}(\boldsymbol{X}_{k+L}^{opt}))) \\ &s.t. \boldsymbol{X}_{k+L}^{opt} = (\boldsymbol{X}_k^{opt\,T}, \cdots, f_v(\boldsymbol{x}_{k+L-1}^{opt}, u_{k+L-1})^T)^T \\ &\quad |u_{k+i}| \leq C, \quad i = 0, \cdots, L-1 \end{aligned} \quad . \tag{8}$$

where C is the limit of the control action.

3.2 Computation of Objective Function for Uncertainty Minimization Task

In Section 3.1, the objective function for uncertainty minimization task is shown using the information matrix of the predicted future pose $\boldsymbol{X}_{k+L}^{opt}$. In this section, based on the assumption of the perfect future measurement $\boldsymbol{Z}_{k+1:k+L}$, we will explain how to obtain this information matrix.

The Fisher information matrix (FIM) of the uncertainty minimization task is computed by:

$$I_{k+L}(\boldsymbol{X}_{k+L}^{opt}) = J(\boldsymbol{X}_{k+L}^{opt})^T \Sigma_{k+L}^{-1} J(\boldsymbol{X}_{k+L}^{opt}), \tag{9}$$

where $J(\boldsymbol{X}_{k+L}^{opt})$ is the Jacobian matrix of the SLAM problem based on the future pose $\boldsymbol{X}_{k+L}^{opt}$, Σ_{k+L} is the diagonal covariance matrix, whose diagonal elements are $\delta_{vx}(k)$ $\delta_{vy}(k)$, $\delta_{\omega}(k)$, $\delta_{fx}(k)$ and $\delta_{fy}(k)$. $\boldsymbol{Z}_{k+1:k+L}$ is a probalistic event. Even though every new perfect measurement will not change the future pose $\boldsymbol{X}_{k+L}^{opt}$, it will greatly change $J(\boldsymbol{X}_{k+L}^{opt})$ and Σ_{k+L}, which leads to the different objective function values.

Firstly, we need to estimate the probalistic $\boldsymbol{Z}_{k+1:k+L}$ based on the future pose $\boldsymbol{X}_{k+L}^{opt}$ and the estimated features $\boldsymbol{x}_{fi}^{opt}$. When the distances between the estimated features $\boldsymbol{x}_{fi}^{opt}$ and the future poses of the robot are smaller than the sensor range, despite the uncertainties, we can assume that this estimated features will introduce the new measurements to the SLAM problem:

$$z_{i,j}^r = \begin{cases} 1 & |\bar{\boldsymbol{x}}_{k+j}^{opt} - \boldsymbol{x}_{fi}^{opt}| \leq R_s \\ 0 & |\bar{\boldsymbol{x}}_{k+j}^{opt} - \boldsymbol{x}_{fi}^{opt}| > R_s \end{cases} , \tag{10}$$

where $z_{i,j}^r$ is the criteria variable to record the belief measurement relationship between the $k+j$-th future pose and the i-th estimated feature. The total number of the probalistic measurements is set as $\sum_{j=1}^{L} \sum_{i=1}^{N_f} z_{i,j}^r = m$, N_f is the number of the estimated features.

Then, $J(\boldsymbol{X}_{k+L}^{opt})$ can be computed by:

$$J(\boldsymbol{X}_{k+L}^{opt}) = \begin{bmatrix} J_r(\boldsymbol{X}_{k+L}^{opt}) \\ J_{add} \end{bmatrix} \tag{11}$$

where $J_r(\boldsymbol{X}_{k+L}^{opt})$ is the Jacobian matrix of $k+L$ vehicle model equations (1) and measurement equations (2) corresponding to $\boldsymbol{Z}_{1:k}$,for the 2D simulation, J_{add} is the added $2m \times (3(k+L) + 2N_f)$ Jacobian matrix. J_{add} is a sparse block matrix. When $\boldsymbol{X}_k^{opt}$ is written as $[\boldsymbol{x}_1^{opt\,T}, \boldsymbol{x}_2^{opt\,T}, \cdots, \boldsymbol{x}_k^{opt\,T}, \boldsymbol{x}_{f1}^{opt\,T}, \cdots, \boldsymbol{x}_{fN_f}^{opt\,T}]^T$, every two ($2*l-1$ and $2*l$, $l = 1, 2, \cdots, m$) rows include two non-zero block (a 2×3 block $\triangle_{i,j}$ and a 2×2 block $\triangle'_j$), where they meet:

$$\begin{aligned} \triangle_{i,j} =& [-R_{k+j}^T, \Gamma R_{k+j}^T(\boldsymbol{x}^{fi} - \boldsymbol{x}_{k+j}^v)] \\ \triangle'_j =& R_{k+j}^T \\ \Gamma =& \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \end{aligned} \tag{12}$$

$\triangle_{i,j}$ locates in $3*k+2N_f+3*j-2$ to $3*k+2N_f+3*j$ columns and $\triangle'_j$ locates in $3*k+2i-1$ to $3*k+2i$ columns. The other elements of J_{add} are zero.

Next, Σ_{k+L} can be obtained by:

$$\Sigma_{k+L} = \begin{bmatrix} \Sigma_{k+L}^r & 0 \\ 0 & \Sigma_{add} \end{bmatrix} \tag{13}$$

where $\Sigma_{k+L}^r = diag(\delta_{vx}(1)^2,\ \delta_{vy}(1)^2,\ \delta_{\omega}(1)^2,\ \cdots,\ \delta_{vx}(k+L)^2,\ \delta_{vy}(k+L)^2,\ \delta_{\omega}(k+L)^2,\ \delta_{fx}(1)^2,\ \delta_{fy}(1)^2,\ \cdots,\ \delta_{fx}(m')^2,\ \delta_{fy}(m'))^2$, m' is the number of known measurements, $\Sigma_{add} = diag(\delta_{fx}(m'+1)^2,\ \delta_{fy}(m'+1)^2,\ \cdots,\ delta_{fx}(m'+m)^2,\ \delta_{fy}(m'+m))^2$.

Finally, we can compute the *D-opt* optimality criterion based on the infromation matrix. In order to avoid overflow and underflow, we can compute the sum of the Log function of the eigenvalues of the information matrix $I_{k+L}(\boldsymbol{X}_{k+L}^{opt})$.

3.3 MPC for Coverage Task under Uncertainty

The area covered by the sensor could be computed from the actual robot trajectory and sensing range. However, the exact robot position is not available but only an estimated position (with uncertainty) is available from SLAM. Thus we need to formulate an expression of the area covered under the localization uncertainty.

Assume at time k, the estimated robot position is $\hat{\mathbf{x}}_k^{opt} \sim \mathcal{N}(\bar{\mathbf{x}}_k^{opt}, I^{-1})$ where $\bar{\mathbf{x}}_k^{opt} = (\bar{x}_k^{opt}, \bar{y}_k^{opt})^T$. The major and minor axes of the 95% confidence ellipse $S_{95\%}$ of a robot position $\hat{\mathbf{x}}_k^{opt}$ is $2\sqrt{5.991}\lambda_1(I^{-1})$ and $2\sqrt{5.991}\lambda_2(I^{-1})$, where $\lambda_1(I^{-1})$ and $\lambda_2(I^{-1})$ are respectively the major and secondary eigenvalues of its covariance matrix. The orientations of the axes of the ellipse are the eigenvectors of the covariance matrix. The range of the sensor is assumed as R_s. If the discrete coordinates of the points in the confidence ellipse $S_{95\%}$ are $(x_i^s, y_i^s), i = 1, ...,$ the coordinates $(x_i^c, y_i^c), i = 1, ...$ of the bound of the covered area S_i^c at the i-th step in the worst case will be (Fig. 1):

$$\begin{aligned} x_i^c &= \frac{l_s - R_s}{l_s}\bar{x}_k^{opt} + \frac{R_s}{l_s}x_i^s \\ y_i^c &= \frac{l_s - R_s}{l_s}\bar{y}_k^{opt} + \frac{R_s}{l_s}y_i^s \quad , \\ l_s &= \sqrt{(\bar{x}_k^{opt} - x_i^s)^2 + (\bar{y}_k^{opt} - y_i^s)^2} \end{aligned} \tag{14}$$

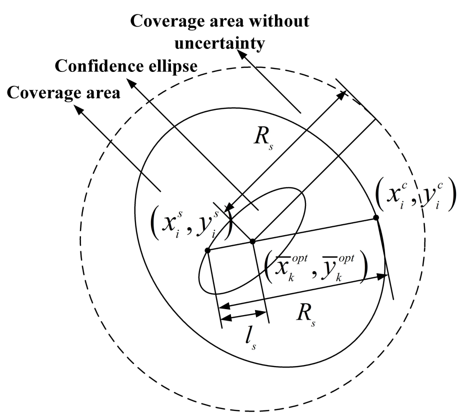

Figure 1: The coverage area under uncertainty

Under the MPC framework, the objective function of the coverage task is defined as:

$$\begin{aligned} f_c(u_{k:k+L_c-1}) &= \begin{cases} \frac{1}{f_A(A_{add})} & A_{add} \neq \varnothing \\ \min\limits_{i=1,\dots,N_r}(l_i^c) & A_{add} = \varnothing \end{cases} \\ A_{add} &= \left(\bigcup_{i=1}^{k+L_c} S_i^c - \bigcup_{i=1}^{k} S_i^c\right) \bigcap Space \quad , \\ l_i^c &= \sum_{j=1}^{L_c} \sqrt{(\bar{x}_{k+j}^{opt} - x_i^r)^2 + (\bar{y}_{k+j}^{opt} - y_i^r)^2} \end{aligned} \tag{15}$$

where $f_A(\star)$ is a function that computes the area of $\star$, L_c is the number of the look-ahead steps, (x_i^r, y_i^r) is the centroid of the i-th uncovered area, N_r is the number of remaining uncovered areas, *Space* means the whole planning space, A_{add} is the new covered space executing L_c look-ahead control inputs, and l_i^c represents the minimum distance between the estimated position of the robot and the centroids of the uncovered areas.

Thus the MPC problem for the coverage task can be formulated as:

$$\begin{aligned} &\min_{u_{k:k+L_c-1}} \quad f_c(u_{k:k+L_c-1}) \\ &s.t. \mathbf{X}_{k+L_c}^{opt} = (\mathbf{X}_k^{opt\,T}, \cdots, f_v(\mathbf{x}_{k+L-1}^{opt}, u_{k+L_c-1})^T)^T. \\ &\quad |u_{k+i}| \leq C, \quad i = 0, \cdots, L_c - 1 \end{aligned} \tag{16}$$

3.4 SQP for Solving MPC Problems

Because the MPC problems for uncertainty minimization and coverage are highly-nonlinear problems, it is difficult to compute its globally optimal solution. We propose to use the SQP approach to compute a sub-optimal result.

3.5 Decision Making for Control Switching

Because the physical meanings and the units of the two objective functions in the two MPC problems are significantly different, it is difficult to find good weights for combining the two objective functions. Thus we proposed to use a switching mechanism using a threshold on the SLAM uncertainty to decide which strategy to implement, either minimizing uncertainty or maximizing coverage. To avoid frequent switching between the two strategies when the uncertainty level is close to the threshold, we propose to use two thresholds instead of one, and the control action will be combined when the uncertainty level is between the two thresholds. The switching scheme is summarized as:

$$\begin{aligned} u_r &= \begin{cases} u_c & Index_1 \leq C_1^{index} \\ c_a u_c + c_c u_a & Index_1 \in (C_1^{index}, C_2^{index}) \\ u_a & Index_1 \geq C_2^{index} \end{cases} \quad , \\ Index_1 &= \sum_{i=1}^{N_f} (\lambda_i^{fx} + \lambda_i^{fy}) \end{aligned} \tag{17}$$

where u_c is the first control input of the coverage task, u_a is the first control input of the *D-opt* MPC solution, and c_a and c_c are weights. C_1^{index} and C_2^{index} are two switching indexes, λ_i^{fx} and λ_i^{fy} are the eigenvalues of the covariance matrices of the features at the x and y axes from SLAM.

4 SLAM based on Linear Submap Joining

In [Huang, 2005] and other earlier research on active SLAM, EKF SLAM is used as the underlying SLAM algorithm. Since there is potential inconsistency in EKF SLAM especially when the orientation error is not very small as in most of the active SLAM scenarios, we proposed to use the optimization based SLAM as shown in (3). It is known that the running-time ability of the optimization based SLAM will decrease if the size of the problem becomes large. Thus we propose to use submap strategy and only perform SLAM, uncertainty minimization and coverage optimization within each local map without the submap joining process, which save a lot of time. We can use some efficient incremental SLAM slovers, such as iSAM2 [Kaess, 2012] and SLAM++ [Viorela, 2017], to finish the SLAM problem in every submap. Whenever we need the global map, we can use submap joining [Zhao, 2013]. Otherwise, the planning in submap is enough.

Two problems arise when using local SLAM and submap joining in active SLAM. The first problem is how to finish the coverage task without having a global map. Here we estimate the robot global pose and its uncertainty without performing the submap joining in every step. We assume that at step k the robot locates at N_i^{sub}-th submap and sub_j-th step of the submap. The estimated coordinate $\boldsymbol{x}_i^{sub_j}$, $i \in \{1,2,...,N_i^{sub}\}$ and its corresponding 2×2 covariance matrix $Cov_i^{sub_j}$ in i-th submap coordinate system can be calculated by:

$$
\begin{aligned}
\boldsymbol{x}_{N_i^{sub}-1}^{sub_j} &= \bar{R}_{N_i^{sub}-1}\boldsymbol{x}_{N_i^{sub}}^{sub_j} + \bar{\boldsymbol{x}}_{N_i^{sub}-1} \\
&\vdots \\
\boldsymbol{x}_1^{sub_j} &= \bar{R}_1\boldsymbol{x}_2^{sub_j} + \bar{\boldsymbol{x}}_1 \\
\boldsymbol{x}_k^{opt} &= \boldsymbol{x}_1^{sub_j} \\
Cov_{N_i^{sub}-1}^{sub_j} &= \bar{R}_{N_i^{sub}-1}Cov_{N_i^{sub}}^{sub_j}\bar{R}_{N_i^{sub}-1}^{-1} + \overline{Cov}_{N_i^{sub}-1}^{sub_j} \\
&\vdots \\
Cov_1^{sub_j} &= \bar{R}_1 Cov_2^{sub_j}\bar{R}_1^{-1} + \overline{Cov}_{\mathbf{1}}^{sub_j} \\
Cov_k^{opt} &= Cov_1^{sub_j}
\end{aligned}
\quad , \qquad (18)
$$

where $\bar{R}_i$ $(i = 1,\cdots,N_i^{sub})$ are respectively the rotation matrices of the last pose of the i-th submap, $\bar{\boldsymbol{x}}_i$ is the xy coordinate of the last pose of the i-th submap, $\overline{Cov}_i^{sub_j}$ is the corresponding 2×2 covariance matrix. So with the current pose in the first local frame $\boldsymbol{x}_k^{opt}$ and its covariance matrix Cov_k^{opt}, we can get the covered area of the robots based on the local maps.

The second problem of using local map in active SLAM is, all the features detected in the new submap will be regarded as new features in the SLAM until the map joining process. This will lead to the robot continuously visiting some old features which have already been mapped in the other submap. To resolve this issue, after starting a new submap, we need to judge whether the feature has been detected and whether its uncertainty has been reduced to an acceptable level in the old submaps or not. If a feature has been observed and its uncertainty is small, it will not be used in the objective function of the uncertainty minimization task.

5 Simulations

In this section, simulation results are shown to validate the effectiveness of the presented algorithm.

5.1 Active SLAM Result using Proposed Method

The simulation environment is created using MATLAB. The robot, using a 20 m limited range omnidirectional sensor, moves in a 50 metre radius non-obstacle circle-shaped space with the known bounds and 30 unknown static features. Specifically, it moves at 1m/s and its control input ω_k is limited in $[-0.3, 0.3]$ rad/s. Synthetic errors, with a Gaussian distribution, are generated for the odometry model of the robot ($\delta_v x(k) = \delta_v y(k) = 0.1$ m and $\delta_\omega(k) = 0.1$ rad) and the sensor measurements ($\delta_f x(k) = \delta_f y(k) = 0.3$ m) is assumed. Three different random features sets are named dataset 1-3. The other parameters are set as: $c_a = 0.85$, $c_c = 0.15$, $C_1^{index} = 0.06N_f$ m, $C_2^{index} = 0.1N_f$ m, the number of the poses of submap is 50. When the whole area is covered more than 97% percent and all features are detected, the simulation will stop. The final results of proposed method using three different datasets are shown in Fig. 2-4.

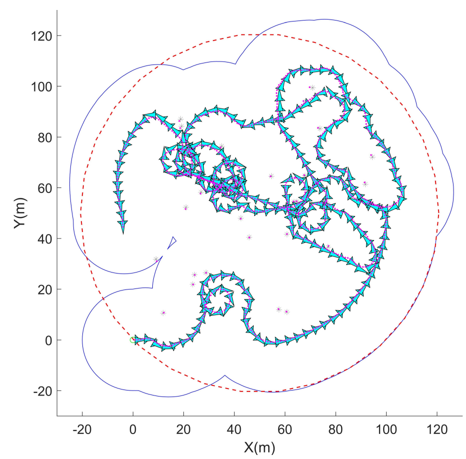

Figure 2: Final results for dataset 1 after using submap joning

In Fig. 2-4, the blue triangles show the real trajectory of the robot at every 5 steps. The purple points show the estimated trajectory and features obtained by SLAM. The black star points are the real positions of the features. Fig. 2-4 show that the robot finishes the SLAM task with good accuracy. Almost all the area of interest (the circle shown by the red dashed line) has been covered (the covered area is shown by the blue curves).

5.2 The Effect of Two Control Inputs

In this part, we will discuss the active control input u_r obtained by two different MPC problems. Fig. 5 shows that the real active control inputs and its coverage rate changes with the simulation time for dataset 1, where > 0, 0 and < 0 respectively mean that the final acting control u_r is the cover-

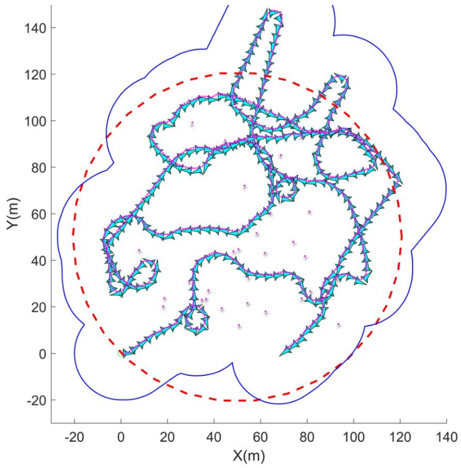

Figure 3: Final results for dataset 2 after using submap joning

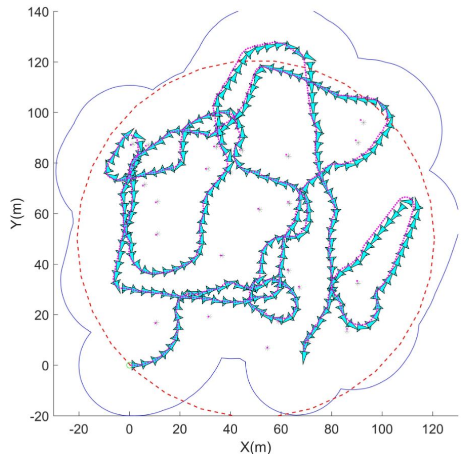

Figure 4: Final results for dataset 3 after using submap joning

age control, combination control or the uncertainty minimization control. We can see that the parameters of the switching mechanism is suitable during the simulation time. Three different control inputs are used rather than the single input.

In the results, we can see that, at the beginning, the uncertainty minimization control input and the coverage control input both can help to increase the coverage rate, because most areas are uncovered. After that, especially after 1000 seconds, most of coverage rate increments are caused by the coverage control input.

5.3 The Effect of Submap Joining Method

The Linear SLAM method can limit the size of the SLAM problem and thus reduce the time for estimation. If we do not use this method, the time used in the SLAM part will become longer and longer while the length of the robot trajectory increases. Here we compare the performance when different submap size is used. Except the submap size, the features and the stop conditions, the other settings are the same as the

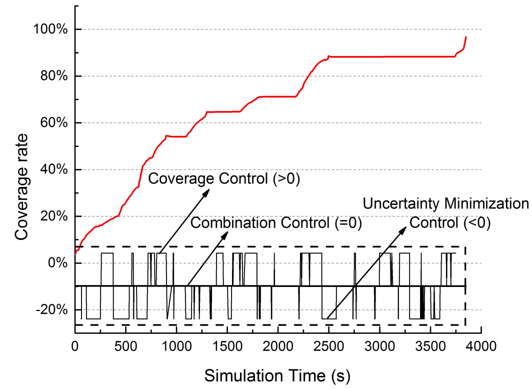

Figure 5: Coverage rate (red curve) and real active control inputs

settings of the simulation described in Section 5.1. Because without using the submap the estimation process will become slow, its stop condition is that the simulation time reaches 5000s. The comparison results are presented in Fig. 6.

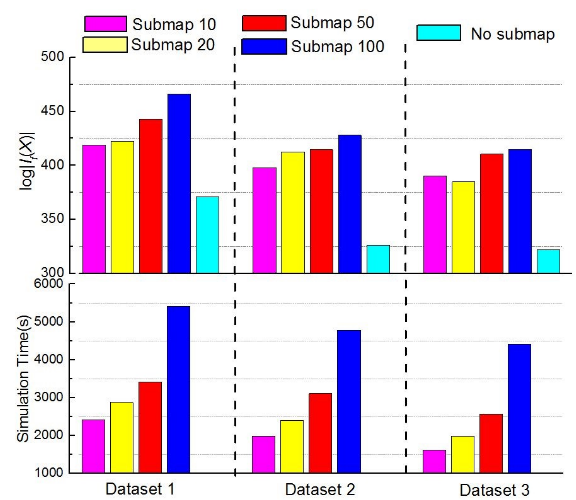

Figure 6: Comparision results using different sizes of submaps (In 5000s, the simulation without using submap does not finish the coverage task. the covered percents from dataset 1 to 3 are respectively 61.6%, 55.6% and 53.2%)

Because of the different number of poses involved, we only compare the information matrix of the features by the *D-opt* optimality criteria $log(det(I_f(X)))$. All simulations using submaps finish the coverage task and detect all features. For the non-submap simulation, because of the slow SLAM process, its total number of measurements (within 5000s) is also less than the ones using submap, which leads to poor SLAM results.

6 Conclusions

In this paper we propose an active SLAM algorithm based on submap joining. We have formulated the *D-opt* uncertainty minimization problem and the coverage problem when pose uncertainty in the submaps is taken into account. By using MPC framework and Linear SLAM for submap joining, the estimation and planning time are significantly reduced. We test our approach in realistic simulation scenarios; experimental results show that the approach is able to deal with the coverage tasks in an unknown environment with a good SLAM result.

In the future, we would like to solve the active SLAM in the complex obstacle environment, taken into account the range of directional sensor. We also plan to extend this algorithm to 3D and perform experiments by an unmanned aerial vehicle platform.

References

[Cadena, 2016] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Scaramuzza Davide, Neira José, Reid Ian, and Leonard John J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, December 2016.

[Huang, 2005] Shoudong Huang, Ngaiming Kwok, Gamini Dissanayake, Quang Ha, and Gu Fang. Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1091–1096, Spain, Barcelona, April 2005. IEEE.

[Kaelbling, 1998] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, January 1998.

[Pazman, 1986] Andrej Pazman, editor. *Foundations of optimum experimental design*. Springer Books, Netherlands, 1986.

[Carrillo, 2012] Henry Carrillo, Ian Reid, and Jose A. Castellanos. On the comparison of uncertainty criteria for active SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2080–2087, USA, MN, Saint Paul, June 2012. IEEE.

[Carrillo, 2015] Henry Carrillo, Yasir Latif, Maria L. Rodriguez-Arevalo, Jose Neira, and Jose A. Castellanos. On the monotonicity of optimality criteria during exploration in active SLAM. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1476–1483, USA, WA, Seattle, May 2015. IEEE.

[Leung, 2006] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898–190, July 2006.

[Indelman, 2015] Vadim Indelman, Luca Carlone, and Frank Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7):849–882, May 2015.

[Indelman, 2015] Vadim Indelman, Luca Carlone, and Frank Dellaert. No correlations involved: decision making under uncertainty in a conservative sparse information space. *IEEE Robotics and Automation Letters*, 1(1):407–414, Jan. 2016.

[Matan, 2014] Keidar Matan and Gal A. Kaminka. Efficient frontier detection for robot exploration. *The International Journal of Robotics Research*, 33(2):215–236 , February 2014.

[Berg, 2012] Van D. Berg, Jur S. Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, September 2012.

[Vallvé, 2015] Joan Vallvé and Juan A. Cetto. Active pose SLAM with RRT*. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1050–4729, USA, WA, Seattle, May 2015. IEEE.

[Maurović, 2017] Ivan Maurović, Marija Seder, Kruno Lenac, and Ivan Petrović. Path planning for active SLAM based on the D* algorithm with negative edge weights. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 99(1):1–11, March 2017.

[Vallvé, 2015] Joan Vallvé and Juan A. Cetto. Potential information fields for mobile robot exploration. *Robotics and Autonomous Systems*, 69:68–79, July 2015.

[Kaess, 2012] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, and John J. Leonard, Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2): 216-235, December 2012.

[Viorela, 2017] Viorela Ila, Lukas Polok, Marek Solony, and Pavel Svoboda. SLAM++-A highly efficient and temporally scalable incremental SLAM framework. *The International Journal of Robotics Research*, 36(2): 210-230, February 2017.

[Zhao, 2013] Liang Zhao, Shoudong Huang, and Gamini Dissanayake. Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 24–30, Japan, Tokyo, January 2014. IEEE.